Doc2Learn Manual

Doc2Learn (Adobe PDF to Learn) was created at the National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign. We would like to acknowledge NARA and NCSA Industrial Partners for the support. The main creators of Doc2Learn are William McFadden, Sang-Chul Lee, Rob Kooper and Peter Bajcsy. This document represents a current description of multiple on-going research and development efforts and hence it is updated on a regular basis.

Introduction

The motivation for developing Doc2Learn (Document to Learn) comes from academic, government and industrial collaborations that involve development of new computer methods and solutions for visual exploration of components in contemporary office documents, such as Adobe Portable Document Format (PDF), and for comparing and grouping sets of documents. Many contemporary documents contain images, text and graphics objects that are complex, heterogeneous in their types, and pose challenges on knowledge extraction. Our objective was to develop a tool that would assist archivists in visual exploration and appraisal of documents for preservation purposes.

The Doc2Learn software is written in Java. It was build as a set of plug-ins to Im2Learn software also developed at NCSA. For more details about Im2Learn, please, visit isda.ncsa.uiuc.edu.

Doc2Learn Functionality

Doc2Learn is designed for visual exploration of heterogeneous information in documents, for performing pair-wise similarity of documents and for grouping of similar documents, as well as for temporal ordering of documents within a group and simple integrity verification of the group of documents. The software computes statistical information about the words, raster images and graphics objects stored in Adobe Portable Document Format (PDF) documents. Documents in other file formats have to be converted into PDF first using polyglot.

The Doc2Learn comparison is a two stage process. The first stage will be executed once and will extract from the PDF document statistics about the digital objects in the document such as word frequency for text, frequency of colors in an image, frequency of encoded vector graphics as well as frequency of samples in an audio file. The second stage of Doc2Learn is to compare multiple documents with each other. Using the extracted statistics a pair-wise comparison is done for all documents selected. The similarities are displayed as a score from 0 (dis-similar) to 1 (the same) and will be color coded for quick viewing. Each of the extracted information for digital objects can have a different weight when comparing (allowing the user to put more weight on text than on images for example).

Installation Instructions

Hardware requirements

For processing large data sets, at least 1GB of RAM is recommended. Depending on the number of parallel jobs executed in the first stage more memory might be required. Depending on the complexity of the document it might require significant computational resources to extract the statistics. However, once the statistics have been extracted the comparison of documents should be only depended on the number of documents that are compared and should be of interactive speeds.

Software requirements

- An Operating System capable of running Java
- Java 1.6

Installation and execution

To install the Doc2Learn application, go to the Image Spatial Data Analysis (ISDA) NCSA website http://isda.ncsa.illinois.edu/download/ and select Doc2Learn from the list of available tools (if not already selected). The installation consists of unzipping the downloaded file into any directory.

The code is executed by double clicking on the scripts according to your Operating System (OS) platform (e.g., .bat file on the Windows OS). On a Mac OS platform and Unix, the scripts needs to be made executable first. This is achieved by using a Terminal, then changing directory to the folder where doc2learn is installed (command "cd"), and finally changing permissions of the scripts by typing "chmod 755 doc2learn-*.sh".

Document Processing

Processing documents can be split in two stages. The first stage can be ran off-line and can be done the moment a document is ingested. Once the document is processed it can be moved to the second stage where the document is compared with other documents. This chapter will go in detail about the two stages of processing.

Stage 1: Statistics Extraction

Extracting statistics from the PDF documents consists of parsing the documents, extracting digital objects, such as text, vector graphics, images, audio, 3D, etc. These digital objects are extracted from the PDF and split up into even smaller pieces, pixels and associated colors in case of images, lines in case of vector graphics, words in case of text, samples in case of audio, vertexes and edges in case of 3D graphics etc. Finally these smallest pieces are counted using histograms. These histograms created are the digital signature of the PDF document.

There are 3 different setups that can be selected from, local, client/server and hadoop. Each of these setups will process the selection of PDF files and will generate the digital signatures of each of the PDF documents. Each of the clients will take the same subset of options: --threads will specify how many threads there need to be and will influence the number PDF documents that can be processed, --output location where the signatures will be stored.

list of PDF files that need to be processed, or folders with PDF files. The local setup, doc2learn-local.bat/sh, will process the PDF files on the local machine. This is used when processing the PDF files on a single large memory machine. The server (as well as the application) will need to have enough memory to hold all the datasets that are processed in parallel (as specified by --threads) in memory. To make sure there is enough memory the bat/sh file needs to be edited and the option -Xmx needs to be modified to change the amount of available memory.

The client/server setup, doc2learn-server.bat/sh and doc2learn-client.bat/sh, will have a queue in the server of all PDF documents that need to be processed and a set of clients that will process all the PDF documents in the queue. This is used when processing data on the HPC systems in NCSA. A single server will be started and clients on each of the nodes in the HPC.

The hadoop setup, doc2learn-hadoop.sh, will upload all the documents to the hadoop filesystem, submit jobs to the hadoop cluster to do the processing, wait for the results to be ready and will retrieve the resulting signatures from the hadoop filesystem. Setting the threads commandline option will allow control over the number of jobs that are submitted to the hadoop queue and are waiting.

Stage 2: Document Comparison

After the signature extraction is completed, which should be run when the data is ingested, the comparison phase begins. Instead of selecting the PDF documents we select the signatures of the documents. Now the comparison is done based on the signatures and is done at runtime.

A tool called doc2learn-viewer.bat/sh will allow the user to select the signatures and look at the comparison. It can show the scores when comparing documents as well as the histograms that where generated during the Stage 1.

Software License

The software is owned by the University of Illinois at Urbana-Champaign.

Developed by: Image Spatial Data Analysis Group (ISDAG) http://isda.ncsa.illinois.edu/

National Center for Supercomputing Applications (NCSA) http://www.ncsa.illinois.edu/

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal with the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimers.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution.
- Neither the names of ISDAG, NCSA or the University of Illinois, nor the names of its contributors may be used to endorse or promote products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.